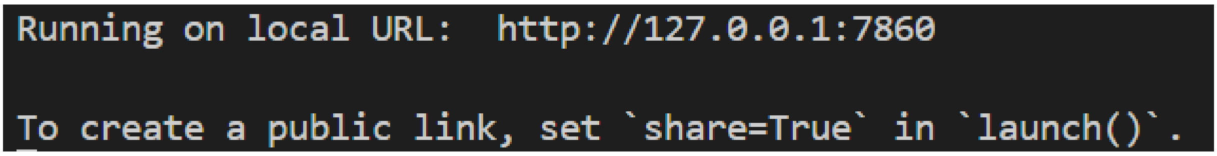
User app documentation

Step 1. Starting the app

After running python main.py on the console, you see in the console something like this :



```
Running on local URL:  http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
```

You click on this link and the app will open in a new browser window.

Step 2. Chunking

a. Chunking explaining

To begin, I want to explain the process of chunking. It has 4 use cases :

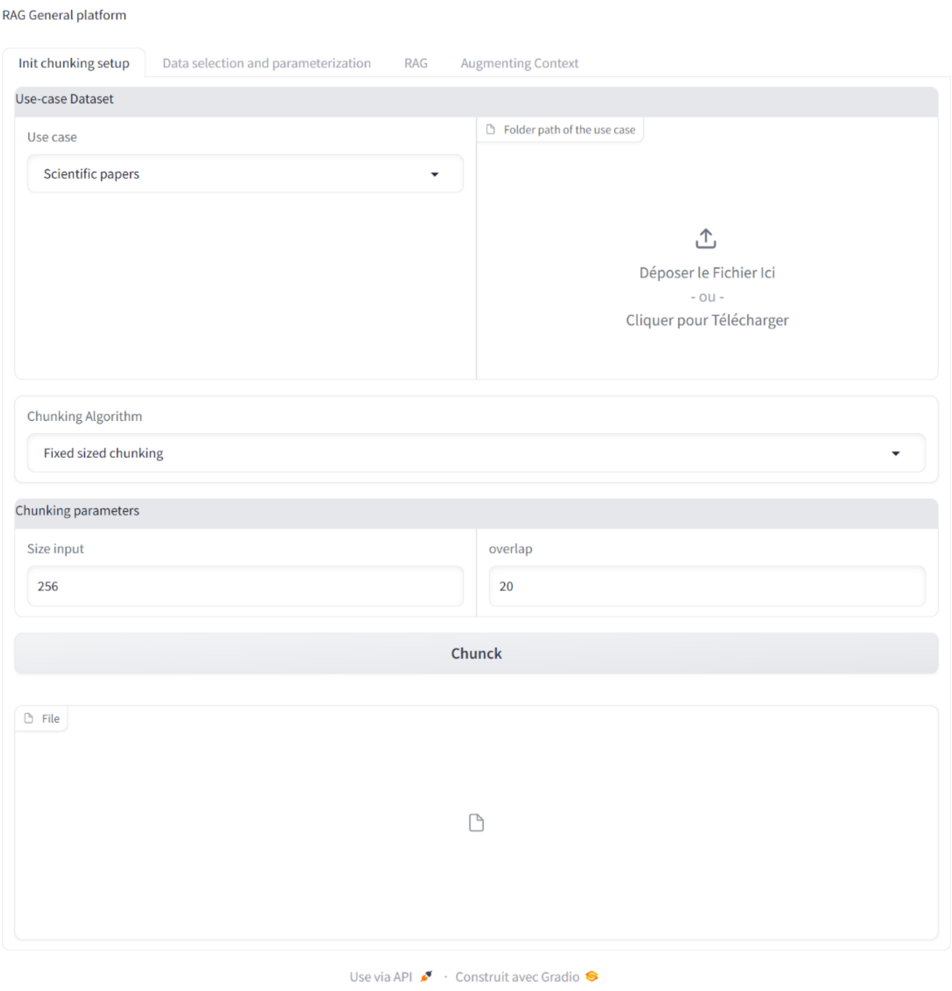
- Scientific papers
- slides and technical reports
- personal docs
- Legal document Analysis

Each chunking use case is saved in a separate folder so that each user can create 4 specific use cases chunking. To do it, the user has to repeat the process of doing one chunking for the 4 use cases, and at the end, he can download the final result which is the downloaded zip file result containing the 4 folders of each use case vector database.

NB: You can do just a few or just one of the 4 use cases, but this time, your use case that is not done is an empty folder in the downloaded zip file result.

b. the process of doing one chunking

This process focuses on just one chunking process. So to begin, the new browser window showing after the click on the link generated in the console before[1] is like this :


RAG General platform

Use-case Dataset

Use case: Scientific papers

Folder path of the use case

Déposer le Fichier Ici
- ou -
Cliquer pour Télécharger

Chunking Algorithm: Fixed sized chunking

Chunking parameters

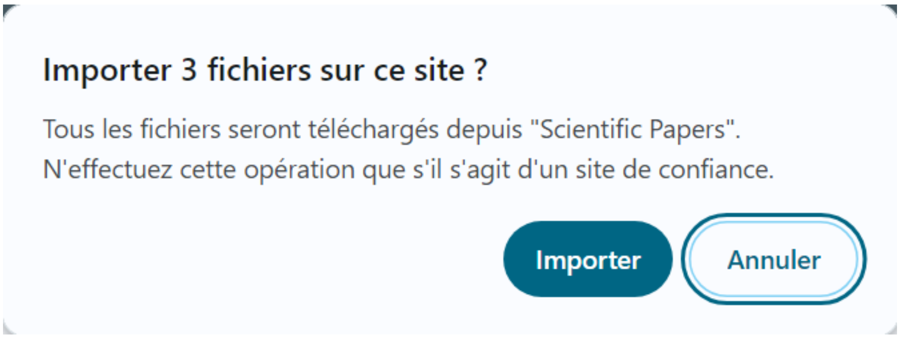
Size input: 256

overlap: 20

You have to fill in all the data in the form apart from the "File" at the bottom. (The "File" at the bottom is for the result and it is unfillable). I also note that the "folder path of the use case" needs a folder.

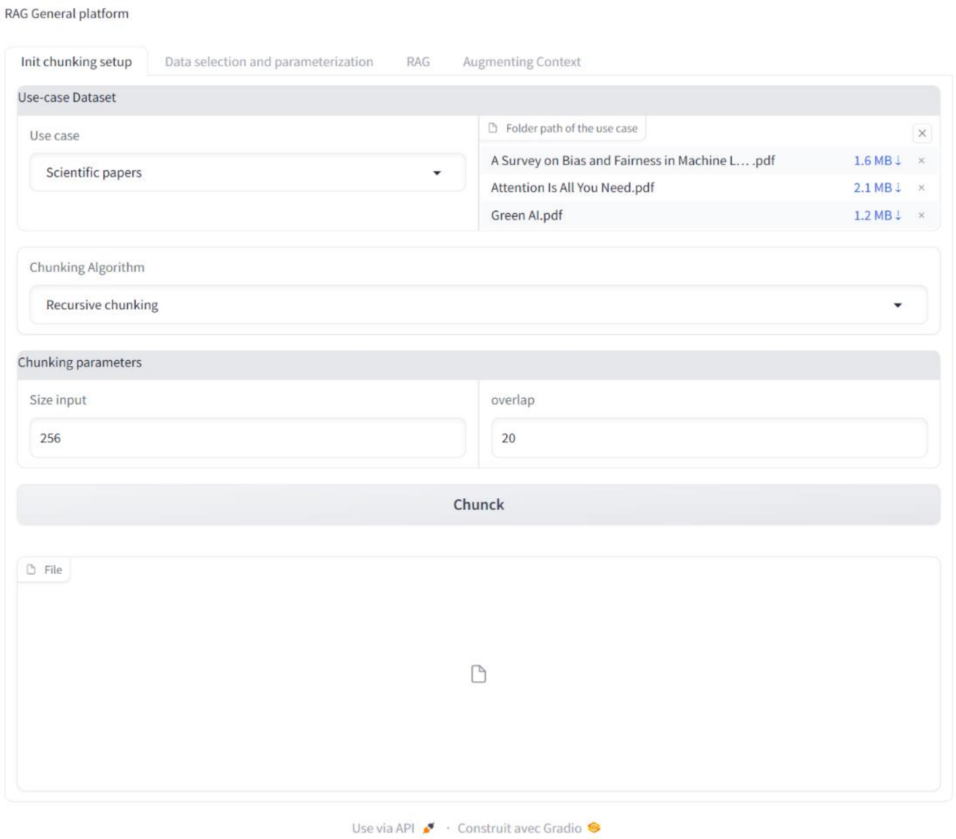
Remark for all the app :

- For all the dropdowns in this app, you have to click on its text to show all the choices (It's specific for Gradio)
- When you upload files in this app, it gives you a popup and you have to click on "import"(importer on French browser) to accept it. The popup is like that :


Importer 3 fichiers sur ce site ?

Tous les fichiers seront téléchargés depuis "Scientific Papers". N'effectuez cette opération que s'il s'agit d'un site de confiance.

When the form is filled it is like this :



Use case: Scientific papers

Folder path of the use case

A Survey on Bias and Fairness in Machine L... .pdf 1.6 MB ↓

Attention Is All You Need.pdf 2.1 MB ↓

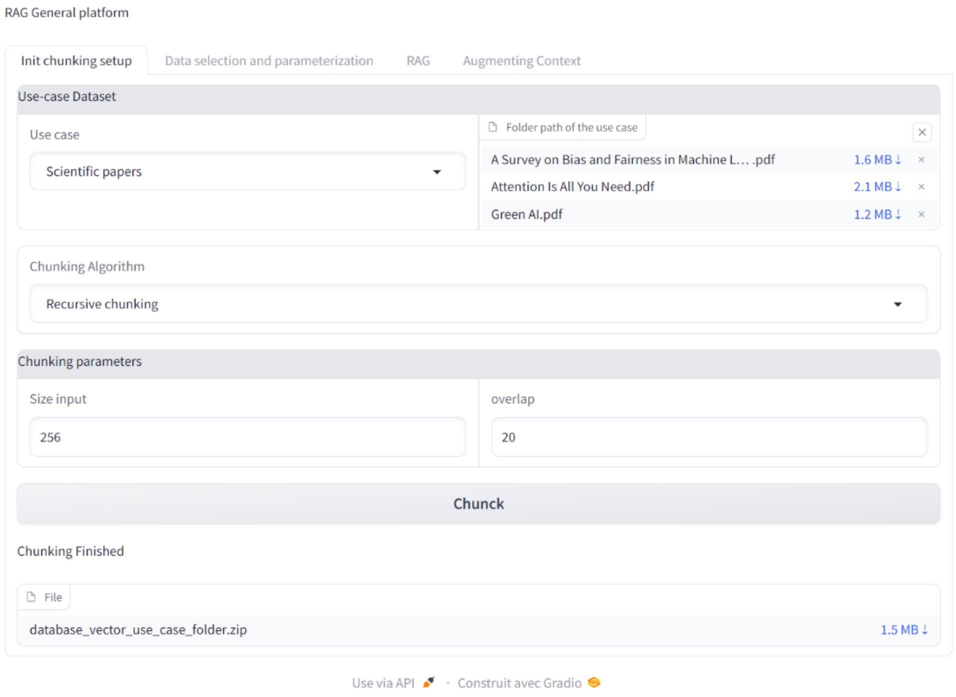
Green AI.pdf 1.2 MB ↓

Chunking Algorithm: Recursive chunking

Size input: 256

overlap: 20

When the button “Chunk” is clicked and the chunking is finished, it is like this :



Use case: Scientific papers

Folder path of the use case

A Survey on Bias and Fairness in Machine L... .pdf 1.6 MB ↓

Attention Is All You Need.pdf 2.1 MB ↓

Green AI.pdf 1.2 MB ↓

Chunking Algorithm: Recursive chunking

Size input: 256

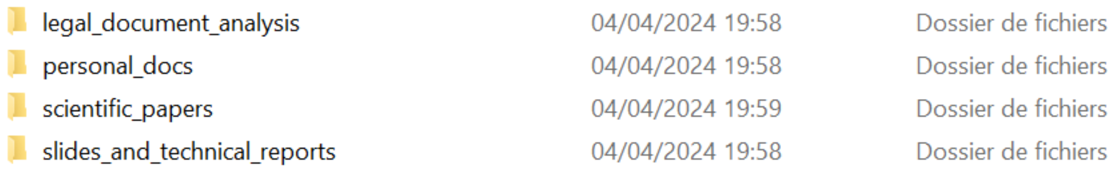
overlap: 20

Chunking Finished

database_vector_use_case_folder.zip 1.5 MB ↓

Now, you can download the zip result file(named database_vector_use_case_folder.zip) in the link in the “File” at the right bottom(in the image the link 1.5 MB↓).

The content of this zip folder is like this :



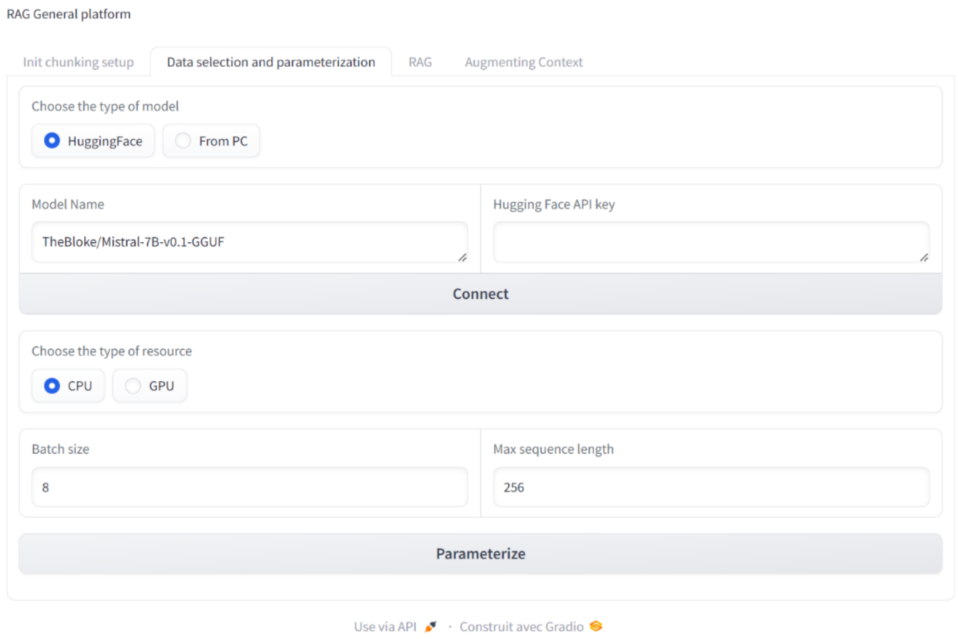
legal_document_analysis	04/04/2024 19:58	Dossier de fichiers
personal_docs	04/04/2024 19:58	Dossier de fichiers
scientific_papers	04/04/2024 19:59	Dossier de fichiers
slides_and_technical_reports	04/04/2024 19:58	Dossier de fichiers

NB: As I explained in the paragraph before, only "the scientific_papers" have content now, so if you want to fill all of these folders. You have to repeat this step for all of the use cases and download the zip folder at the end.

Step 3. Parameterization

a. Initial state

This tab is for the parameterization of the model to be used for the RAG. This tab at the beginning is like this :



Choose the type of model

● HuggingFace ○ From PC

Model Name: TheBloke/Mistral-7B-v0.1-GGUF

Hugging Face API key:

Choose the type of resource

● CPU ○ GPU

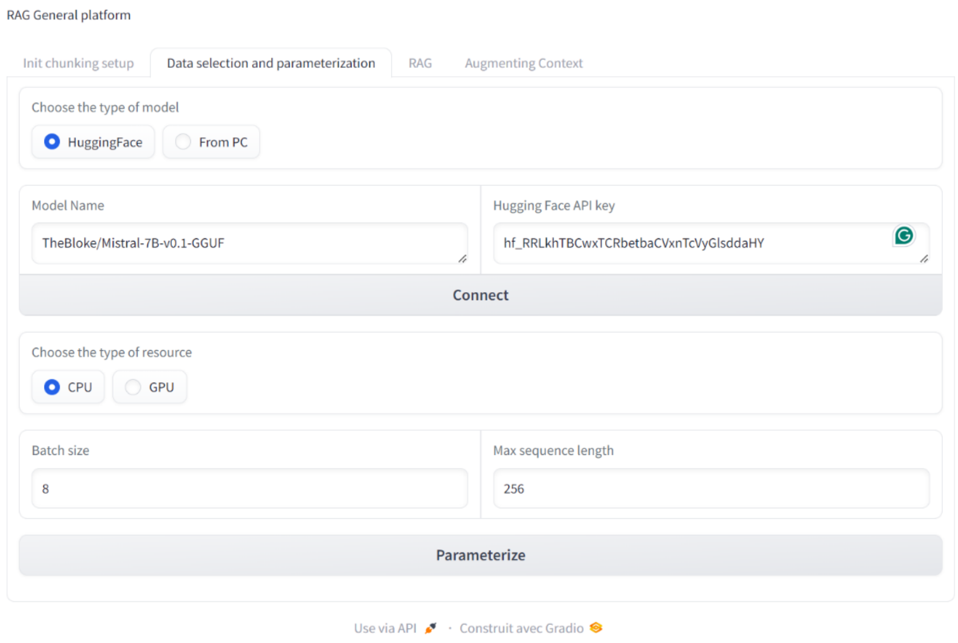
Batch size: 8

Max sequence length: 256

b. Hugging Face parameterization

i. Test of connection

To begin the hugging face parameterization, we can test the connection by filling in the model name and the Hugging Face API key like this :



Choose the type of model
● HuggingFace ○ From PC

Model Name: TheBloke/Mistral-7B-v0.1-GGUF

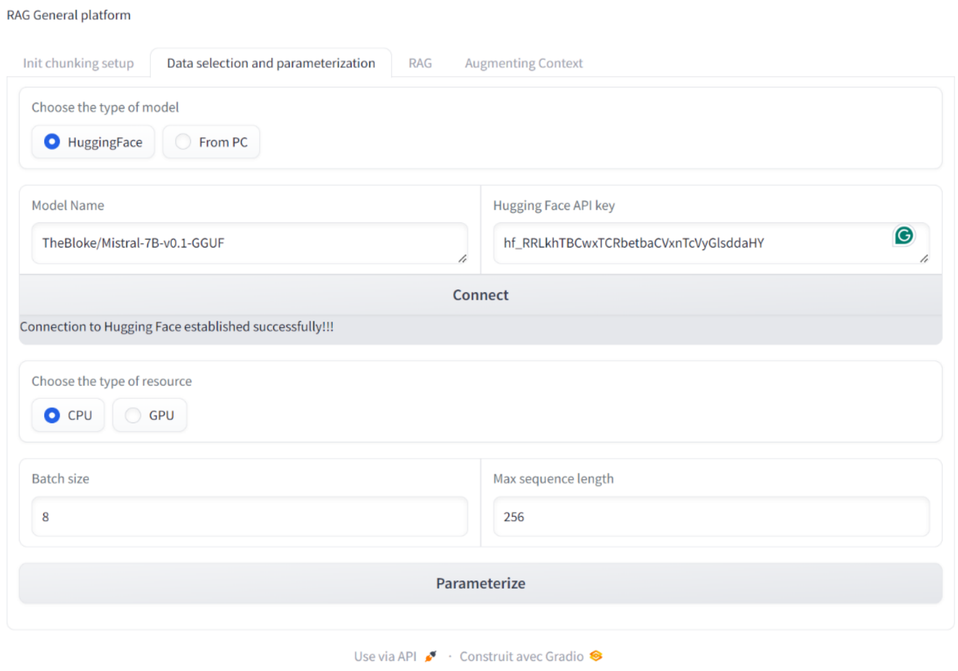
Hugging Face API key: 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

Choose the type of resource
● CPU ○ GPU

Batch size: 8

Max sequence length: 256

When we click on the button connect, the result is like this:



Choose the type of model
● HuggingFace ○ From PC

Model Name: TheBloke/Mistral-7B-v0.1-GGUF

Hugging Face API key: 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

Connection to Hugging Face established successfully!!!

Choose the type of resource
● CPU ○ GPU

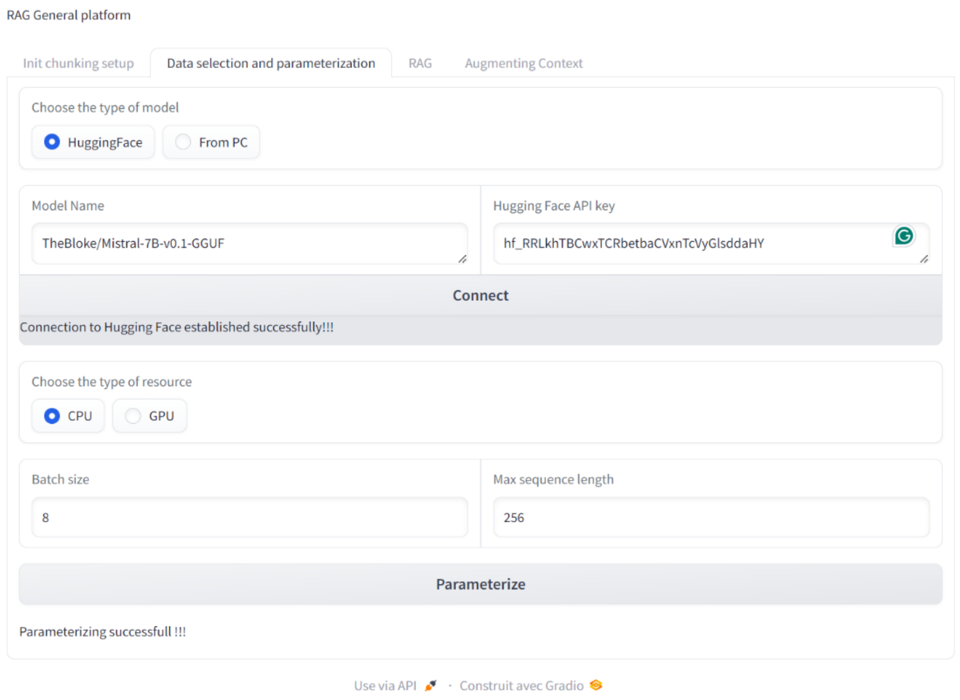
Batch size: 8

Max sequence length: 256

We see in our case that the connection is successful with the message "Connection to Hugging Face established successfully!!!". If the connection has failed, we have seen a message as "Failed to establish connection to Hugging Face"

ii. Doing parameterization :

Now that the connection is established, we can now click the button "Parameterize". And the result is like this :



Choose the type of model: ● HuggingFace ○ From PC

Model Name: TheBloke/Mistral-7B-v0.1-GGUF

Hugging Face API key: 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

Connection to Hugging Face established successfully!!!

Choose the type of resource: ● CPU ○ GPU

Batch size: 8

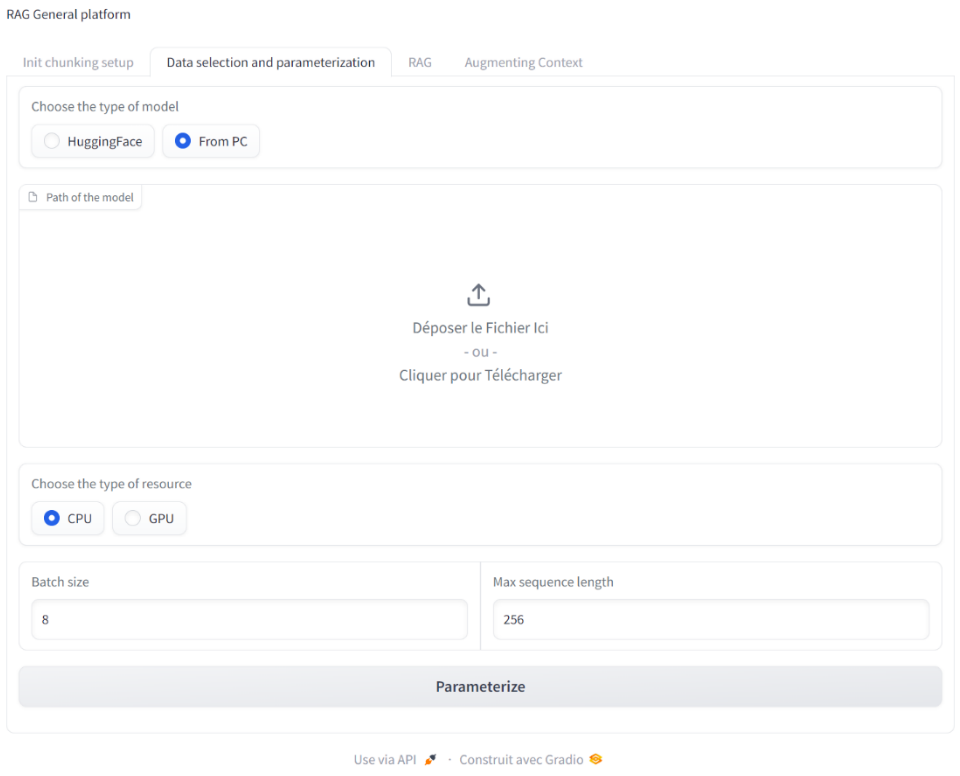
Max sequence length: 256

Parameterizing successfull !!!

We see the message "Parameterizing successful" that confirms the parameterization.

c. Model File From PC parameterization

When your model is a file from a PC? You chose the type of model "From PC" and you see this:



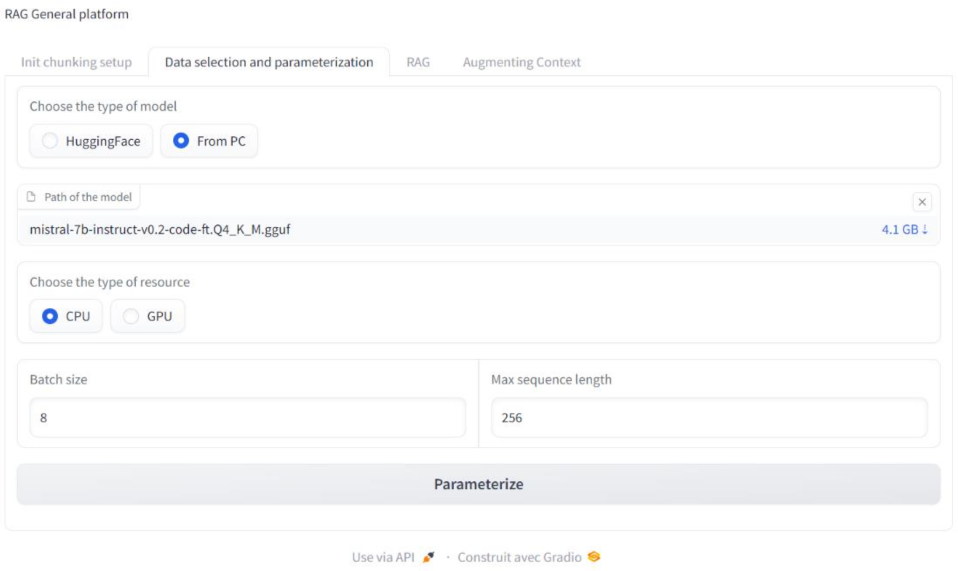
Choose the type of model: ○ HuggingFace ● From PC

Choose the type of resource: ● CPU ○ GPU

Batch size: 8

Max sequence length: 256

After filling the model file(only '.bin' and '.gguf' is accepted), and the other parameters, the tab is like this :



RAG General platform

Init chunking setup | Data selection and parameterization | RAG | Augmenting Context

Choose the type of model

○ HuggingFace ● From PC

Path of the model ×

mistral-7b-instruct-v0.2-code-ft.Q4_K_M.gguf 4.1 GB ↓

Choose the type of resource

● CPU ○ GPU

Batch size: 8

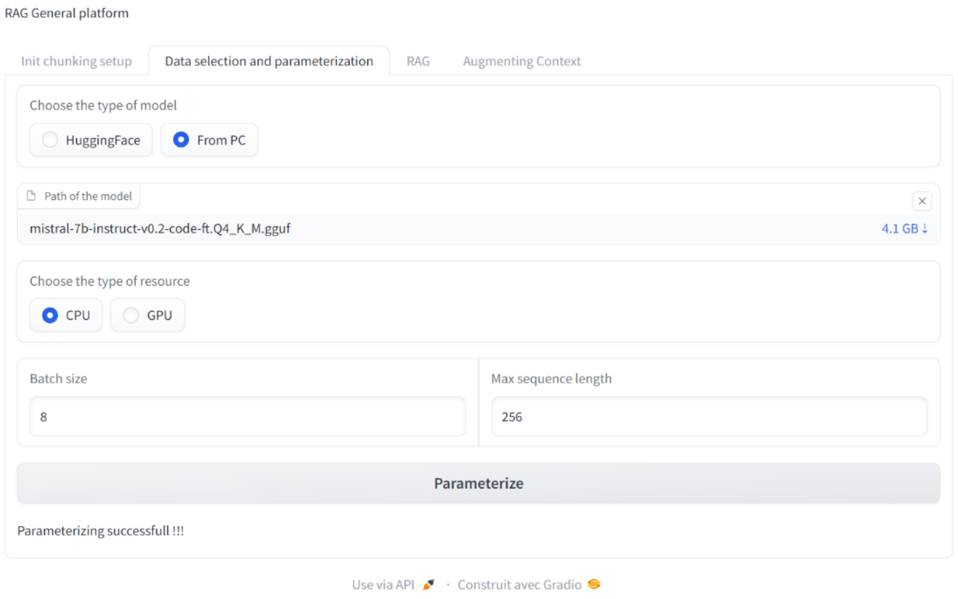
Max sequence length: 256

Parameterize

Use via API · Construit avec Gradio

NB: If you want to change the model, you have to click the button"x", and upload the new model.

After clicking the button "parameterize", the tab is like this :



RAG General platform

Init chunking setup | Data selection and parameterization | RAG | Augmenting Context

Choose the type of model

○ HuggingFace ● From PC

Path of the model ×

mistral-7b-instruct-v0.2-code-ft.Q4_K_M.gguf 4.1 GB ↓

Choose the type of resource

● CPU ○ GPU

Batch size: 8

Max sequence length: 256

Parameterize

Parameterizing successfull !!!

Use via API · Construit avec Gradio

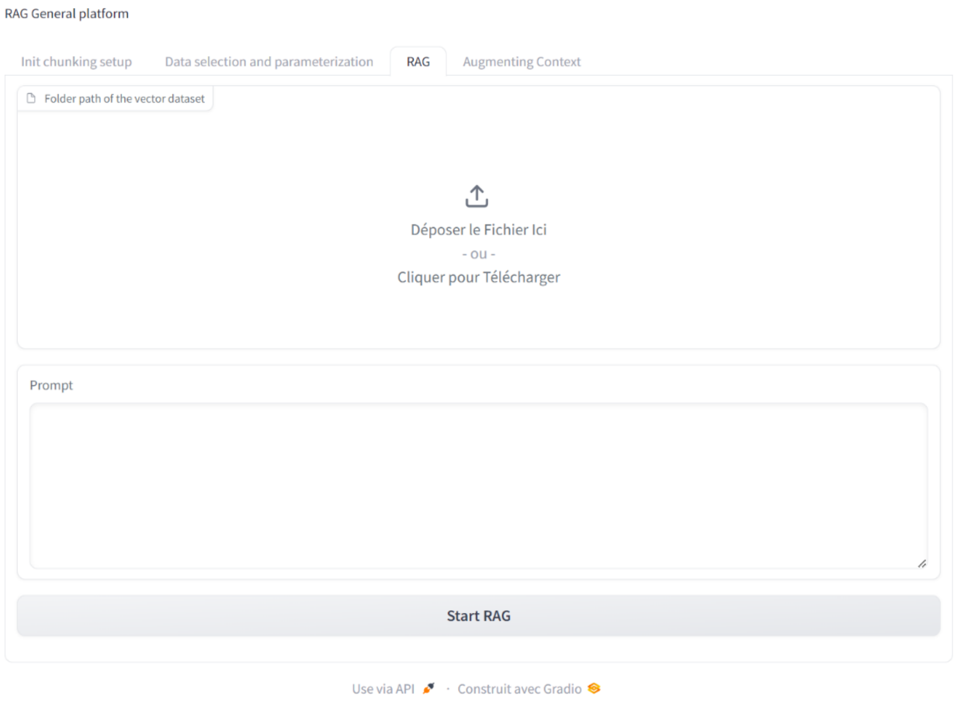
We can see the message "Parameterizing successful" that confirms the parameterization.

NB: If you choose GPU, ensure that your machine has a Nvidia GPU that is compatible with Cuda.

Step 4. Doing RAG

a. Initial state

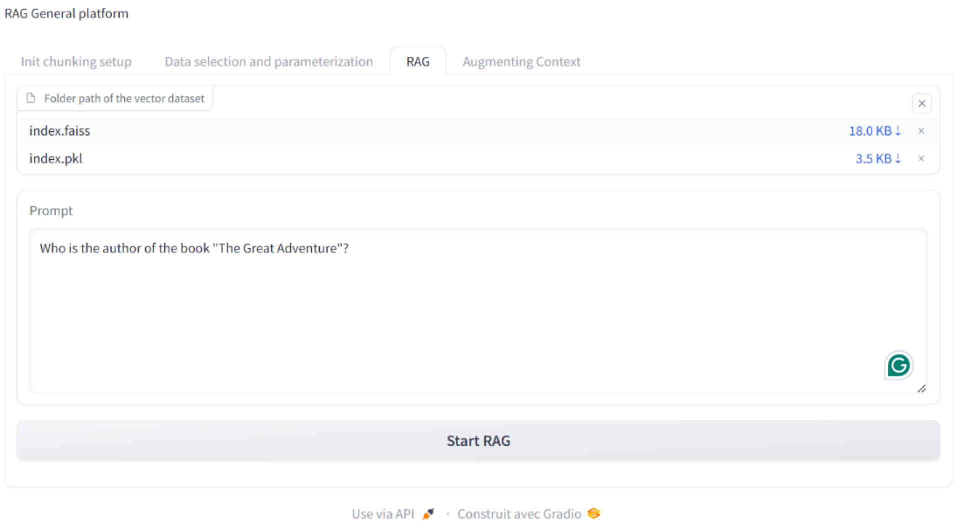
In this step, we are doing RAG, so the initial state of the rag tab is this :



We need to provide the folder of the database vector and the prompt text of the RAG.

b. Filling the RAG parameters

After filling the folder of the database vector and the prompt text of the RAG, the tab is changed like this :



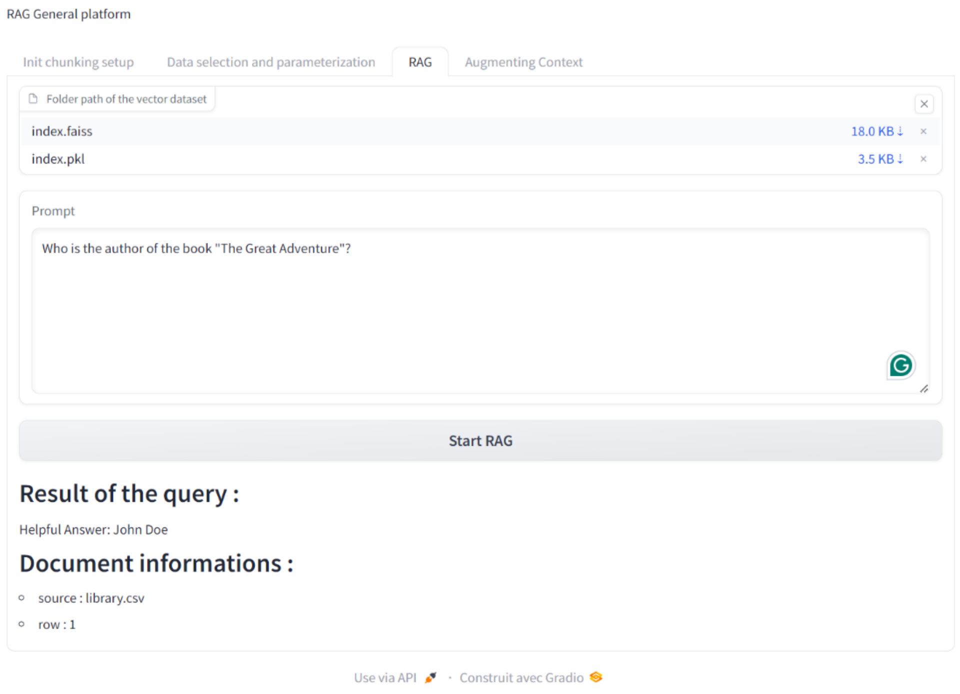
index.faiss 18.0 KB ↓

index.pkl 3.5 KB ↓

Who is the author of the book "The Great Adventure"?

c. The RAG result

After clicking the button "Start RAG" and when the RAG generation is finished, we see the result like this :


RAG General platform
Init chunking setup
Data selection and parameterization
RAG
Augmenting Context
Start RAG
Use via API · Construit avec Gradio

Folder path of the vector dataset

index.faiss 18.0 KB

index.pkl 3.5 KB

Prompt

Who is the author of the book "The Great Adventure"?

Result of the query :

Helpful Answer: John Doe

Document informations :

- source : library.csv
- row : 1

As a result, we see two things:

- Result of the query
- Document informations

In our case, the result of the query "Who is the author of the book "The Great Adventure" ?" is "John Doe", and it is from the document "library.csv" in row 1 of the data.

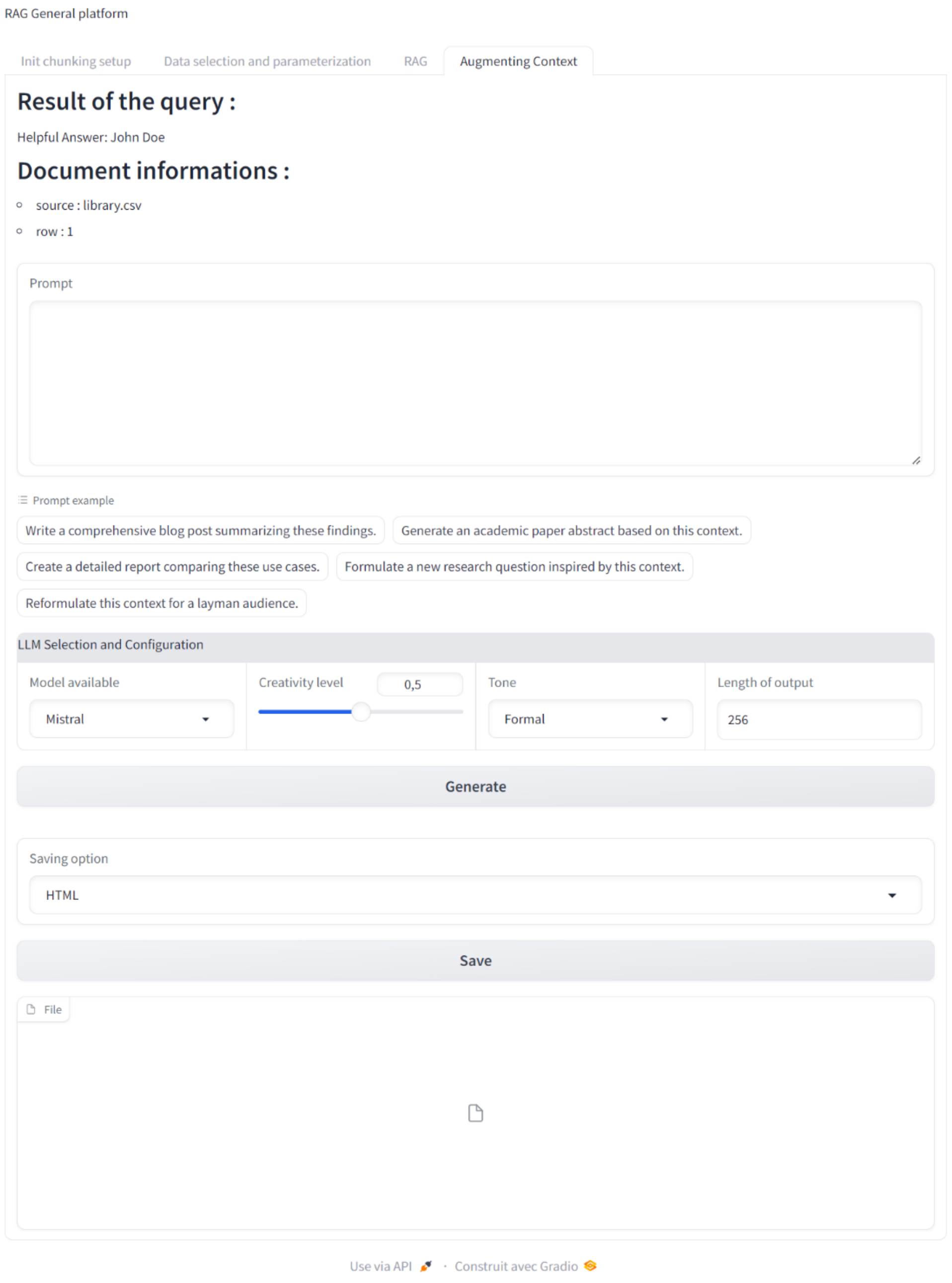
NB:

- The document information may vary from one document to another
- For the database vector that utilizes "Agentic chunking", the document informations is empty because the processing data of "Agentic chunking" is very different than the other methods of chunking.
- At the first RAG generation, we must provide the "Folder path of the vector dataset"
- For the second and the other generation, and if the "Folder path of the vector dataset" is always the same, you can delete the content of the "Folder path of the vector dataset" to accelerate the RAG process but it is optional

Step 5. Augmenting Context

a. Initial state

This new tab is for augmenting the context to ameliorate the result of the RAG. So when we go in this tab (after a RAG process in the previous tab), we see the result of the query and the documents information before and the prompt area and some LLM selection and configuration like this :


RAG General platform
Init chunking setup
Data selection and parameterization
RAG
Augmenting Context

Result of the query :

Helpful Answer: John Doe

Document informations :

- source : library.csv
- row : 1

Prompt

Prompt example
Write a comprehensive blog post summarizing these findings.
Generate an academic paper abstract based on this context.
Create a detailed report comparing these use cases.
Formulate a new research question inspired by this context.
Reformulate this context for a layman audience.

LLM Selection and Configuration

Model available: Mistral

Creativity level: 0,5

Tone: Formal

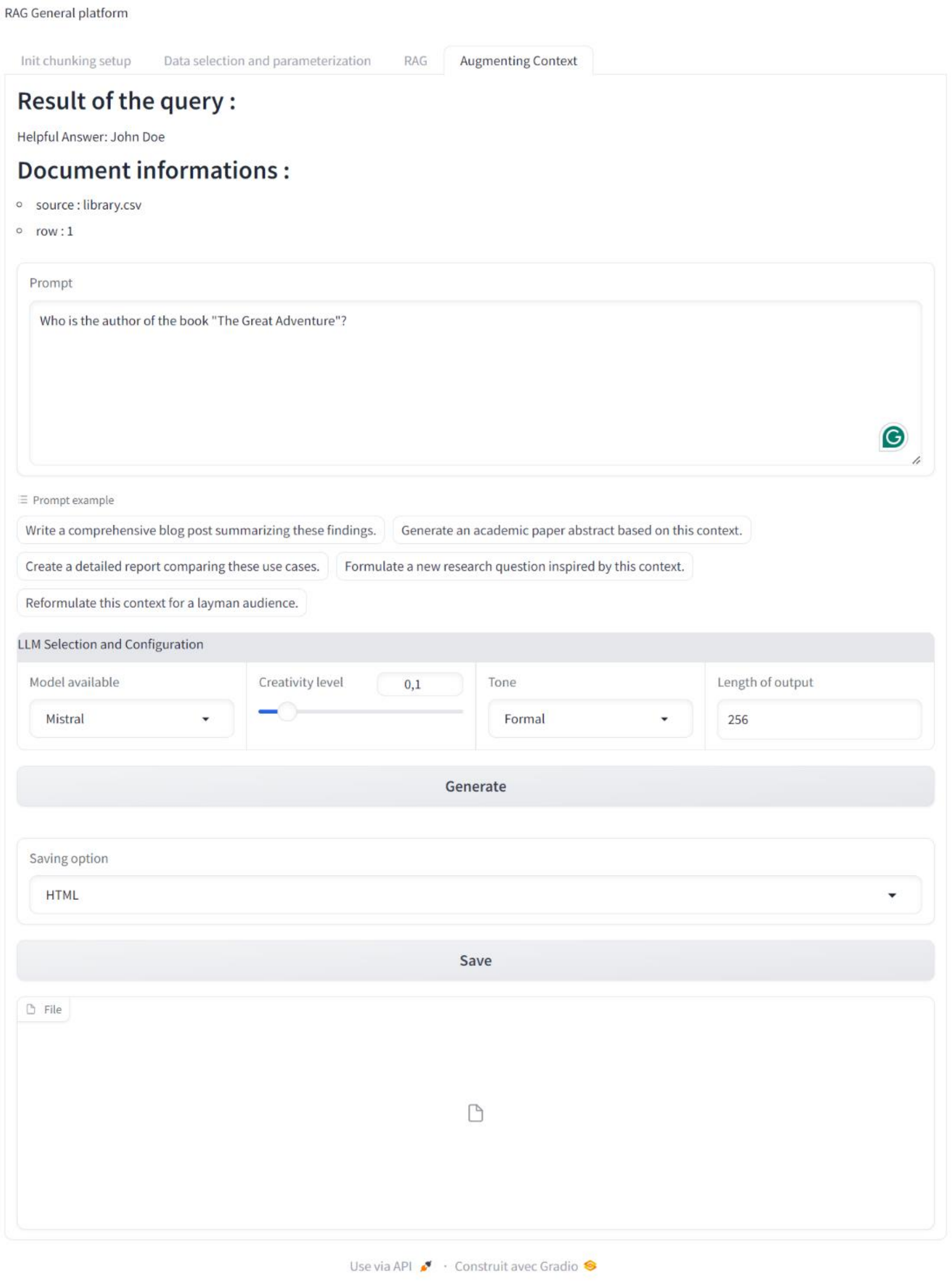
Length of output: 256

Saving option: HTML

Save
File
Use via API · Construit avec Gradio

b. Filling the information

You can ameliorate the prompt with the prompt area that you can fill. In addition, this app provides you some examples of prompts that are already in the “Prompt example”, if you want to use this, you just click on it and the prompt is automatically filled in the prompt area. For the LLM Selection, you can choose between “Mistral” and “Llama2”. For the create level it is from 0 to 1. For the tone, you have 40 tones available. For the length of the output, you can fill in the number that corresponds the you.



RAG General platform

Result of the query :

Helpful Answer: John Doe

Document informations :

- source : library.csv
- row : 1

Prompt: Who is the author of the book "The Great Adventure"?

Prompt example

- Write a comprehensive blog post summarizing these findings.
- Generate an academic paper abstract based on this context.
- Create a detailed report comparing these use cases.
- Formulate a new research question inspired by this context.
- Reformulate this context for a layman audience.

LLM Selection and Configuration

Model available: Mistral

Creativity level: 0,1

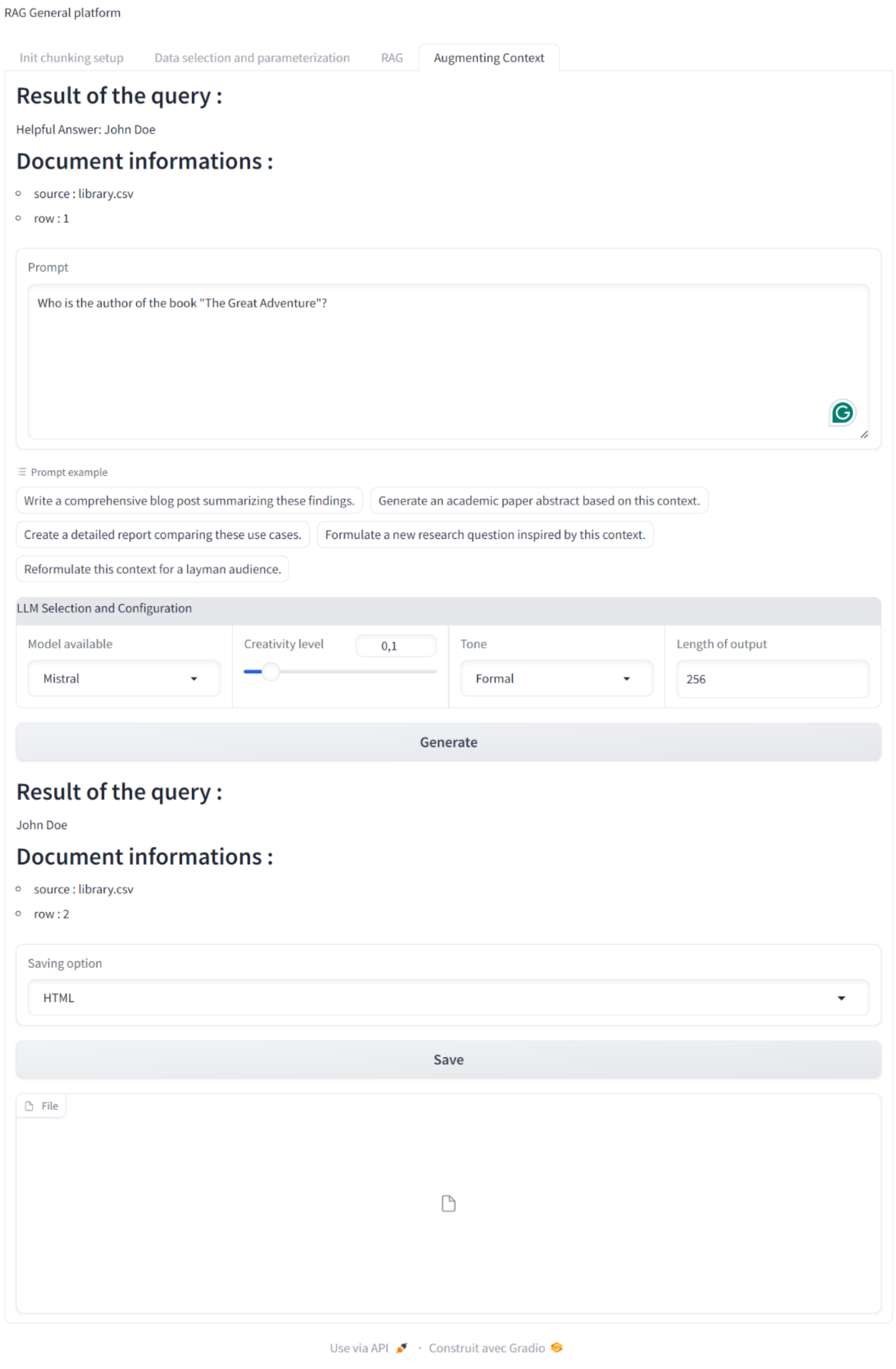
Tone: Formal

Length of output: 256

Saving option: HTML

c. Result of the augmented context

After clinking the button generate and after the process is finished, the tab is changed like this :



RAG General platform

Init chunking setup
Data selection and parameterization
RAG
Augmenting Context

Result of the query :

Helpful Answer: John Doe

Document informations :

- source : library.csv
- row : 1

Prompt

Who is the author of the book "The Great Adventure"?

Prompt example
Write a comprehensive blog post summarizing these findings.
Generate an academic paper abstract based on this context.
Create a detailed report comparing these use cases.
Formulate a new research question inspired by this context.
Reformulate this context for a layman audience.
LLM Selection and Configuration

Model available: Mistral

Creativity level: 0,1

Tone: Formal

Length of output: 256

Result of the query :

John Doe

Document informations :

- source : library.csv
- row : 2

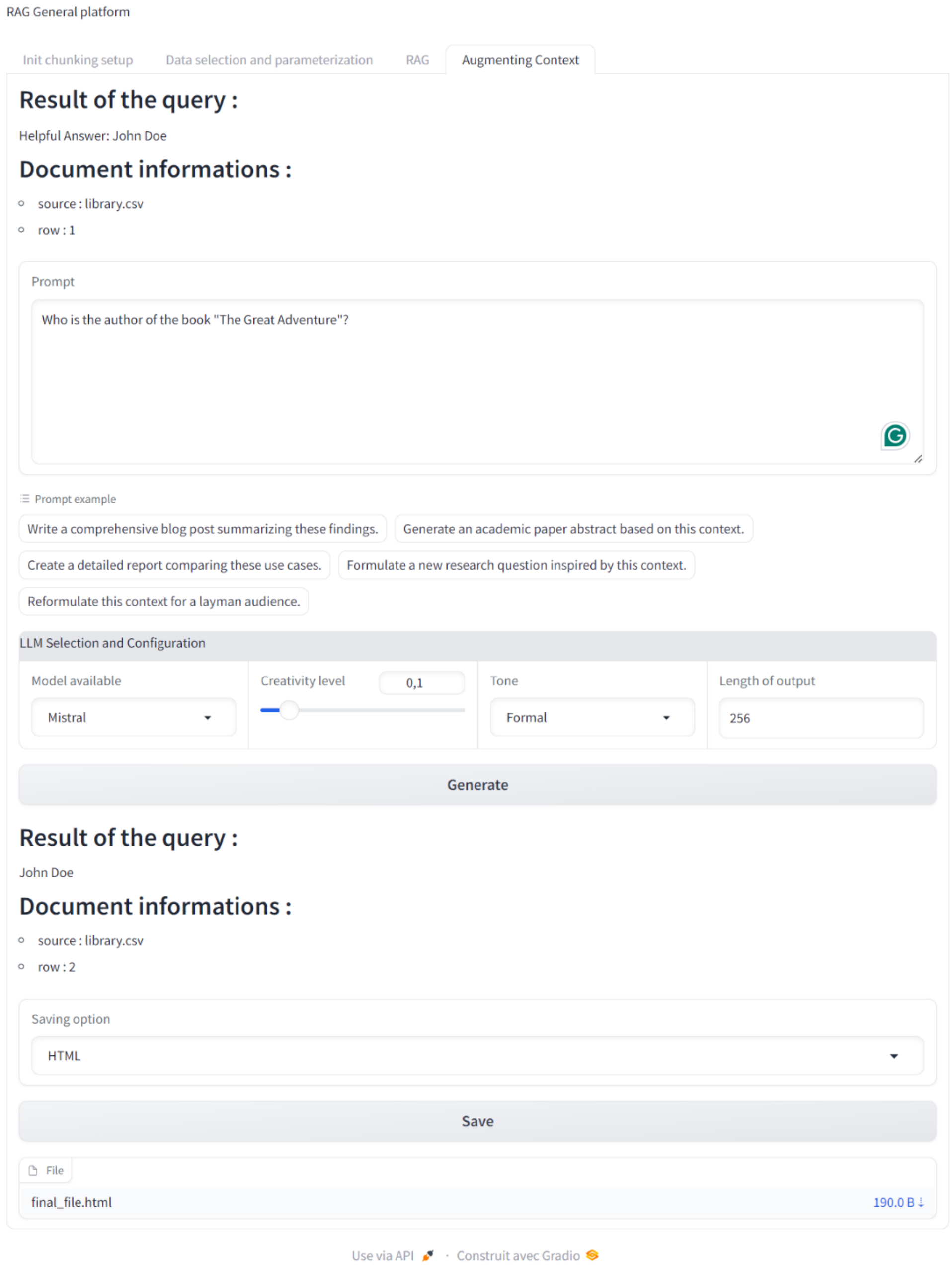
Saving option: HTML

Save
File
Use via API
Construit avec Gradio

You can save this final result in different formats (DOCX, PDF, HTML). You can make this choice by clicking the text in the dropdown menu (in our image the text HTML) and all of the saving options are shown. After that, click on the button “Save” to generate the RAG final result save file.

d. Download result final file

After generating the final RAG final result file, the tab is like this:



Result of the query :

Helpful Answer: John Doe

Document informations :

- source : library.csv
- row : 1

Prompt

Who is the author of the book "The Great Adventure"?

Prompt example

- Write a comprehensive blog post summarizing these findings.
- Generate an academic paper abstract based on this context.
- Create a detailed report comparing these use cases.
- Formulate a new research question inspired by this context.
- Reformulate this context for a layman audience.

LLM Selection and Configuration

Model available	Creativity level	Tone	Length of output
Mistral	0,1	Formal	256

Result of the query :

John Doe

Document informations :

- source : library.csv
- row : 2

Saving option

HTML

final_file.html 190.0 B ↓

You can see, that our final result is in the file “final_file.html” that we can download in the right bottom link (in our case, the link 190.0B↓).